DOCUMENT RESUME

ED 324 217 SE 051 596

AUTHOR Funkhouser, Charles
TITLE Mathematical Problem Solving: A Review of the Literature.
PUB DATE 90
NOTE 43p.
PUB TYPE Information Analyses (070)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS Algorithms; Artificial Intelligence; *Cognitive Structures; Computation; *Computer Assisted Instruction; Computers; Critical Thinking; Elementary Secondary Education; Learning Strategies; Literature Reviews; Mathematics Education; *Memory; Misconceptions; Physics; *Problem Solving; Skill Development

ABSTRACT

The major perspectives on problem solving of the twentieth century are reviewed--associationism, Gestalt psychology, and cognitive science. The results of the review on teaching problem solving and the uses of computers to teach problem solving are included. Four major issues related to the teaching of problem solving are discussed: (1) translation training; (2) schema training; (3) strategy training; and (4) algorithm automaticity. The use of computers and problem solving, problem solving environments, the use of artificial intelligence in problem solving, and the impact of computer programming on problem solving are considered. Included is a bibliography of 71 references. (KR)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED324217

MATHEMATICAL PROBLEM SOLVING: A REVIEW OF THE LITERATURE

Dr. Charles Funkhouser
California State University at San Bernardino

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Charles Funkhouser

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☑ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

SE051596

MATHEMATICAL PROBLEM SOLVING: A REVIEW OF THE LITERATURE

Literature on Problem Solving in General

Definitions of Problem Solving

The substance of problem solving is problems. Jerome Bruner (1961) often cited a light-hearted definition of a problem by a British philosopher named Weldon:

> [He] asserted there are three basic things one should consider in defining what a problem is: troubles, puzzles, and problems. A trouble is a circumstance, a situation, that leaves one upset and at loose ends; one knows that things aren't going very well. It is a source of vague, but persistent discomfort. A puzzle has a nice tight form, clear structure, and a neat solution. A problem, then, is what you get if you can find a puzzle form to lay on top of a trouble. (p.439)

Rucinski (1986) noted a similarly whimsical definition of problem solving suggested by Bernard Corman (1958). In Corman's review of literature on problem solving, Corman said one could "consider problem solving as meaning the behaviors that researchers who say they are studying problem solving, study" (p.459).

Throughout this century, mathematics educators have been interested in problem solving. In the Second Yearbook of the National Council of Teachers of Mathematics, the causes of failure in problem solving were discussed (NCTM, 1926). The psychology of problem solving was investigated

in the NCTM's Third Yearbook (NCTM, 1928). In its "Agenda for Action," the NCTM (1980) stated that "the acquisition of problem solving skills should be one of the goals of mathematics instruction in the 1980's." (p. 3) It was listed on the agenda as its first recommendation.

What have been some of the major perspectives on problem solving during this century? This review of the literature will look at some of the ways researchers who have studied such behaviors have talked about problem solving. These ways are varied, but tend toward three perspectives: associationism, Gestalt psychology, and cognitive science. A review of literature related to these perspectives follows.

Associationism

According to Mayer (1983), associationism can be traced back to three laws of learning and memory postulated by Aristotle:

1. Doctrine of association by contiguity--Events or objects that occur in the same time or space are associated in memory, so that thinking of one will cause thinking of the other.
2. Doctrine of association by similarity- Events or objects that are similar tend to be associated in memory.
3. Doctrine of association by contrast--Events or objects that are opposites tend to be associated in memory (p.11).

Edward Thorndike (1911) helped popularize a theory of learning following from these laws called associationism. Thorndike placed a cat into what he called a "puzzle box." If the cat performed a certain behavior, a door would open allowing the cat to get food. His theory of human learning was derived from these animal experiments.

According to Thorndike, the first step in finding a solution to a problem is a series of random guesses. Some of these guesses lead to a successful solution, others do not. The solver tries many responses until the problem is solved. Thorndike found that as practice increased, the solver had a tendency to perform more often those responses that did work, and to perform less often those that did not work. That is, successful behaviors tended to be repeated, unsuccessful behaviors tended to be extinguished. He called this learning by trial and error or "accidental success." In short, Thorndike's associationism explained present behaviors according to associations built up in the past (Hilgard, Atkinson, & Atkinson, 1971).

Gestalt Theory

Kohler (1925) presented problems similar to Thorndike's to apes and reached different conclusions. He presented food to an ape requiring the ape to use a stool and stick to reach it. Based on his observations, Kohler suggested animals solved problems by a "flash of insight." The animals first thought about the problem, then found a solution when everything fell into place. Kohler's

interpretation of problem solving is called a Gestalt theory of learning; that is, a problem situation is considered and a solution is found by sudden insight.

Gestalt psychologists, such as Kohler (1925) and Wertheimer (1959) defined the process of problem solving as a search to relate one aspect of a problem to another. They called this search "structural understanding." The solver must organize the elements of a problem until they fit together. In the end, the solver achieves a flash of insight to reach a solution.

Polya (1957, 1965) attempted to describe the components of the Gestalt concepts of structure, organization, and insight in a more rigorous way. He postulated a series of steps in solving a problem:

1. Understanding the problem: The solver gathers information about the problem.

2. Devising a plan: The solver uses past experiences to find a method of solution.

3. Carrying out the plan: The solver tries out the plan of solution, checking each step.

4. Looking back: The solver tries to check the result by using another method or by seeing how the parts fit together.

Subsequent theorists, such as Kendler and Kendler (1962) and Underwood (1965), attempted to reconcile associationism and Gestaltism by describing problem solving as a process of mediation. According to mediationists,

associationists believed thinking could be represented by a stimulus (S) going in and a response (R) coming out. The problem solving event could be represented as (S-R). Mediationists suggest that the S evokes a miniature internal response called a mediational response, r_m. The r_m creates a new internal state, or s_m, which may evoke an additional mediational response, r_m, with the process repeating until the solution response, R, is reached. This may be represented as:

$$S - r_{m1} - s_{m1} - r_{m2} - s_{m2} - \ldots - r_{mn} - s_{mn} - R.$$

The capital S is the initial problem situation, the capital R is the solution, and the lower case s's and r's are the mediational states and responses. Mediationism suggested the "flash of insight" is not a sudden, spontaneous phenomenon, but the result of a series of intermediate steps.

Cognitive Science

Cognitive theorists, such as Bartlett (1932), Mayer (1983) and Davis (1984), also postulate problem solving as a multiple step process. Bartlett (1932) defined schemata as the "active organization of past reactions which must always be supposed to be operating in any well-adapted organic response." (p. 201) He defined assimilation as the search for the appropriate "setting" of past experiences. He stated:

> Whenever such settings are found, facts of "meaning" emerge ... all the cognitive processes, from perceiving

> to thinking are ways in which some fundamental effort after meaning seeks expression. Speaking very broadly, such effort is simply the attempt to connect something that is given with something other than itself. (p. 227)

As commonly presented in the literature, the cognitive science concepts of schemata and assimilation can be thought of as following from Gestaltist and mediational theories of problem solving.

Mayer (1983), like Corman (1957), saw a lack of agreement within the literature on a definition of problem solving. He suggested the terms problem solving, thinking, and cognition are used interchangeably. But in considering the major work in problem solving, he found three basic characteristics common to definitions of problem solving. The charactistics were:

1. [Problem solving] is cognitive, but is inferred from behavior.
2. [It] is a process that involves some manipulation of or set of operations on knowledge in the cognitive system.
3. [It] is directed and results in behavior that "solves" a problem or is directed toward a solution. (p. 7)

According to Mayer (1983), cognitive theory is very useful in talking about problem solving. Cognitive theory "involves finding how a present problem relates to those

concepts and ideas that already exist in a problem solver's memory; that is, external relations between elements and schemata." (p.68) The problem must be assimilated into the individual's own experience and translated into terms familiar to the individual.

From the cognitive science perspective, problem solving is the process of figuring out which set of past experiences (schema) best relate to the problem at hand. The problem solver must then interpret the new situation based on the schema selected. Mayer (1983) suggested the concept of schemata is more clearly defined if looked at in four ways:

1. General: A schema may be used in a wide variety of situations as a framework for understanding incoming information.

2. Knowledge: A schema exists in memory as something a person knows.

3. Structure: A schema is organized around a theme.

4. Comprehension: A schema contains "slots" that are filled by specific information within a broad framework. This process of "filling in the slots" or interpreting the new situation into selected schema is called assimilation.

In Davis' model, for example, problem solving consists of two simultaneous processes: problem representation and analysis of the representation. Correct problem solution depends on the appropriateness of the representation and analysis. Davis further subdivided these two processes into seven subprocesses. These subprocesses are:

1. Examining the problem for cues or clues to guide retrieval.
2. Retrieving from memory some knowledge representation structure that seems likely to be helpful.
3. Gradually building up a representation for the problem.
4. Building up a representation for a matching piece of "knowledge."
5. Making two mappings, the first of which maps the problem (as input data) into its representation, and the second of which maps the problem representation into the "knowledge" representation.
6. Evaluating the adequacy of the constructions, retrievals and mappings, and accepting or rejecting portions of this work.
7. Cycling back through steps 1 through 6, as often as necessary. (p. 366)

A number of educators see a unique opportunity for problem solving in mathematics education because of cognitive science. Resnick and Ford (1981) expressed interest in applying cognitive science principles to research in mathematical problem solving. They saw the work of researchers in cognitive science and mathematics education as complementary.

> As psychologists concerned specifically with mathematics, our goal is to ask the same questions that experimental developmental psychologists ask about learning, thinking, and intelligence but to focus

> these questions with respect to a particular subject matter. What this means is that instead of asking ourselves, "How is it that people think?", we ask ourselves, "How is it that people think about mathematics?". (p. 3)

Schoenfeld (1982) expressed similar hope for cooperation between cognitive scientists and mathematics educators:

> The mathematics education community cannot afford to ignore the psychological research on problem solving. But it cannot afford to swallow it whole, either. Mathematics educators, I think, have had their hearts in the right place but have lacked methodological tools that allowed for substantive and rigorous inquiries into problem solving. (p. 35)

Mayer (1985) also saw cognitive science as a natural partner of mathematics educators researching problem solving. He suggested cognitive science can supply the methodological tools for analyzing problem solving, while mathematics can contribute problem solving tasks and opportunities for research. Like Schoenfeld (1982), he was concerned for the use of quantitative measures in conducting such research. As a cognitive scientist and educator, Mayer suggested four approaches to teaching problem solving:

1. Translation training: The solver must be able to translate each sentence of the problem into an internal representation. He cited research which suggests difficulty

with linguistic comprehension is a source of difficulty in problem solving.

2. Schema training: Alluding to Gestalt theory, he noted the need for a solver to be able to put the elements of a problem into a coherent whole. Research by Hinsley, Hayes, and Simon (1977) and Silver (1979) suggested that lack of knowledge of problem types is a critical source of difficulty in problem solving.

3. Strategy training: The solver needs to be trained in ways to solve problems. Research by Bloom and Broder (1950), Rubinstein (1981) and Schoenfeld (1979) suggested that it may be possible to train solvers on "how to" solve problems.

4. Algorithm automaticity: This last area concerns itself with the use of algorithms in problem solving. Research by Brown and Burton (1978) suggested errors in problem solving can often be traced to systematic errors in solvers' computational algorithms. Research by Case (1978) and Davis (1984) also indicated the development of sophisticated algorithms requires that simpler algorithms first become automatic for the solver.

Lastly, Kilpatrick (1985) also suggested that cognitive science models may be useful for research into mathematical problem solving. He stated:

> The most heartening developments of the past few years reside not in the tangible contributions of specific research studies but in changing approaches to

> research. The converging interest of cognitive scientists and mathematics educators ... can only strengthen future research efforts in both camps. And the emerging view of the mathematics teacher as a full partner in the research process, rather than an imperfect system for delivering instruction, can only yield research that will better mediate between theory and practice. (p.13)

This review of definitions of problem solving suggest the most useful definitions are associationist, Gestaltist, and cognitive science theories of problem solving and learning. Definitions, treatment, and instrumentation can be chosen after reflection on these theories. The author recognizes, however, that the process of problem solving may lie outside the scope of these theories. Continued research may suggest other useful ways of looking at the problem solving process.

Literature on Teaching Problem Solving

As previously noted, Mayer (1983) saw four major issues from cognitive science which are related to the teaching of problem solving. They are summarized as follows:

1. Translation training: comprehension versus solution
2. Schema training: understanding versus executing
3. Strategy training: process versus product
4. Algorithm automaticity: procedures versus response

What follows is a discussion of the literature related to

the teaching of problem solving centered around these four issues.

Translation Training

Mayer (1983) suggested that to solve a problem one must be able first to translate the problem into an internal representation. He also suggested current instruction in problem solving often ignores representation of problems and concentrates on solutions instead.

Greeno (1980) noted the difficulty learners have in comprehending relational sentences in problems. For example, elementary students in arithmetic were given the problem, "Joe has three marbles. Tom has five more marbles than Joe. How many marbles does Tom have?" This proposition was sometimes repeated by the student as "Joe has three marbles. Tom has five marbles. How many marbles does Tom have?"

Soloway, Lockhead and Clement (1982) asked students to write equations to represent relational sentences such as "There are 10 times as many students as teachers at this school." Given this relational sentence, about a third of the students produced equations like 10S = T.

In a similar study, Mayer (1982b) asked college students to read a series of eight algebra story problems, then to restate them. They often restated them in confused form. It appeared students tended to lack the linguistic skill for translating relational sentences into internal

representations. Questions for possible research in problem solving would include:

1. Will students who are given practice involving training in representing relational propositions show improved performance in problem solving?

2. Are there common errors learners make in translating relational sentences which could be guarded against during problem solving instruction?

Schema Training

The previous section discussed the literature on translation of problems into internal representation. This section discusses how studies suggest the representation must be put into a meaningful form to reach a solution.

Paige and Simon (1966) sought to demonstrate that solving a problem requires more than sentence translation. They presented problems, such as "The number of quarters a man has is seven times the number of dimes he has. The value of the dimes exceeds the value of the quarters by $2.50. How many has he of each coin?"

Students translated the sentence into equations, such as 7Q = 7D and D(.10) = 2.50 + Q(.25). Some students, however, were able to see that the problem was not solvable as given.

Other researchers have investigated a learner's ability to recognize the category to which a problem belongs. Hinsley, et al. (1977) gave subjects a series of algebra

problems from textbooks and asked them to organize the problems into categories. Students generally were able to perform this task correctly. In fact, students tended to be able to select the correct category after reading only the first few words of the problem. In a follow-up study, Hayes, Waterman, and Robinson (1977) found students also could use their categorization of the problem as a "schema" or "frame" to select relevant information and algorithms for the solution.

Recent work by Mayer (1981, 1982a) produced similar results. He developed a taxonomy of basic problem types. General categories, such as "motion problems," were subdivided into different sub-classes, such as "overtake," "round trip," and "speed change." He saw several implications for teaching problem solving following from studies of schema training. Among these was the need to provide sufficient practice in a variety of problem types to avoid misapplication of different schemata by the student.

In a more recent study, Mayer (1982b) asked students to read and then recall a series of eight story problems. Students tended to recall high-frequency versions of problems more easily than low-frequency versions. In addition, there was a tendency of students to turn low-frequency versions of problems into high-frequency versions. This suggests when students lack enough practice in selecting an appropriate schema for a problem, the representation is more prone to error.

Riley (1982) did a follow-up study to Greeno (1980) and Mayer's (1983) work on the identification of problem types by students. He identified three types of word problems:

1. Cause/change: "Joe has 3 marbles. Tom gives him 5 more marbles. How many marbles does Joe have now?"

2. Combination: "Joe has 3 marbles. Tom has 5 marbles. How many marbles do they have together?"

3. Comparison: "Joe has 3 marbles. Tom has 5 more marbles than Tom. How many marbles does Tom have?"

Although these problems each require the same computation (3 + 5), Riley found substantial differences in performance on different problem types. Children in grades K and 1 performed well on cause/change problems, but poorly on comparison problems. Children in grades 2 and 3 performed well on all types. This suggested a developmental trend in which children begin with the idea that all problems are cause/change problems, and later begin to differentiate among different problem types. Errors in problem solution may be due to students' lack of appropriate schemata.

Additional questions for research include the following:

1. Will problem solving performance increase if students are given practice in recognizing problem types?

2. Is recognition training enhanced when exercises contain a mixture of problem types rather than having all problems solvable by the same procedure?

3. Will problem solving performance improve if students are given practice in representing problems, where forms of representation may be concrete, pictorial, symbolic, or verbal?

4. Will problem solving performance improve if students are given practice in selecting relevant and irrelevant information in a problem?

Strategy Training

"Process" in problem solving refers to the strategy used to reach an answer. "Product" in problem solving refers to the answer obtained. The literature suggests that much of the current instruction in problem solving involves product rather than process. The literature further suggests that concern for instruction in process rather than in product may be more productive in the teaching of problem solving.

At the University of Chicago, Bloom and Broder (1950) studied a group of students who were having difficulty passing comprehensive examinations in various subjects. These students scored high in scholastic aptitude and claimed to study conscientiously. This group was called the "remedial group," while a group of similar students who were able to pass their comprehensive exams without difficulty was called the "model group." Model students were asked about the strategies they used when solving problems. These strategies were compiled. Remedial students were taught to

imitate these strategies. Students who had this training tended to score significantly higher on the examinations than matched students who did not have the strategy training.

In more recent work, Rubinstein offered a course at the University of California at Los Angeles entitled, "Patterns of Problem Solving." This course, offered since 1969, focused on the problem solving process and taught students the application of common strategies. Reif (1981) claimed no objective data are available on Rubinstein's work, but a report by Rubinstein (1981) offered numerous student testimonials about the course's value.

Schoenfeld (1979) reported similar results in his study involving the teaching of problem solving strategies. All subjects received written and tape recorded instruction on how to solve 20 similar problems. The experimental group also was given a description of five useful strategies, such as "draw a diagram," "break it into smaller parts," or "consider a similar problem." Subjects took a five problem pre- and posttest. The experimental group showed strong pre- to posttest gains. It should be noted, however, that there were few subjects in this study, making the strong gains in test scores not as convincing.

In a similar study, Chisko and Davis (1986) trained college students to solve "practical" problems. For example, students were to act as consumer advocates. Students composed a letter to a television manufacturing

company to accomplish the goal of satifying a customer whose television set had caught fire. Another problem, called the "Community Cable Problem," required the students to draw a design for wiring cables to buildings for a cable television company. Chisko and Davis trained the students in problem solving skills in a series of workshops which concentrated on recognizing types of problems, defining problems, organizing information, modeling, analyzing data, flexibility, and generalizing and consolidating. Their results indicated students made strongest gains when they were actively involved in practicing these specific problem solving skills. They also reported positive affective gains for these same students with regard to problem solving.

Greeno (1978) developed a computer model to teach heuristics for geometry proofs. The project, PERDIX, taught "propositions of inference," such as vertical angles are congruent, and "concepts," such as the recognition of vertical angles. The project also included what Greeno called "strategic knowledge," the setting of goals and forming of solution plans. Though no specific data were cited to support his claims, he reported that such strategic knowledge could be taught to enhance problem solving ability.

To conclude, the literature suggests three areas of strategy training which seem worth further study:

1. How can worked-out examples be used to teach problem solving strategies?

2. How can direct instruction and practice in using specific strategies be used to enhance performance?

3. Does having students systematically describe and compare their solution procedures enhance problem solving performance?

Algorithm Automaticity

To conclude this review of the literature related to teaching problem solving, one last group of skills will be discussed. Mayer (1983) called these skills "component skills", since they are the components used to develop the algorithms in problem solving.

An example of a component skill is computational skill. A traditional view of the acquisition of computational skill was offered by Thorndike (1922). Consistent with the associationist theory of learning discussed earlier, Thorndike saw computational skill in arithmetic as the acquisition of stimulus-response bonds. As Mayer noted, one shortcoming of this view is the enormous memory required for complex computations. Mayer and others suggested an alternative view which sees component skills requiring the acquisition of algorithms, not simply a stimulus-response association.

Groen and Parkman (1972) are among those with an alternative view. They developed several models for children's learning of the addition algorithm. Three of these models are:

1. Counting-all: Set a counter to 0. Increment it by m and then by n. For 3 + 4, the child counts 1, 2, 3, 4, 5, 6, 7.

2. Counting-on: Set a counter to the first number, m, then increment by the second, n. For 3 + 4, the child would count 4, 5, 6, 7.

3. Choice: Set a counter to the larger of m or n, then increment by the smaller. For 3 + 4, the child would count 5, 6, 7.

They found that most first graders used the choice model for their addition algorithm. This suggests that instruction in addition should foster the correct application of the choice model on appropriate problems.

In contrast to Groen and Parkman (1972), Fuson (1982) found most pre-schoolers applied the "counting all" algorithm for addition, then developed other strategies as they gained more experience with computation. Such procedures Fuson called "known facts" and "derived facts." Known facts are the memorized results of simple arithmetic facts. Derived facts are developed from known facts. For example, if a student knows 6 + 6 = 12, he can derive 5 + 7 = 12 by taking 1 from the 7 and giving it to the 5. This derived fact could be developed by induction after working similar problems or sufficient practice with manipulatives. Known facts and derived facts are consistent with the associationist view of learning; that is, present knowledge is based on knowledge from the past.

Fuson (1982) suggested that the progression of learning computational skill begins with the Groen and Parkman (1972) counting-all, counting-on, and choice models, then progresses to known facts, then to derived facts. He did not suggest, however, how computational skill might develop from there to more sophisticated algorithms. He also relied heavily on observational data from a small number of subjects.

Woods, Resnick and Groen (1975) suggested a comparable model for subtraction. The algorithms students tend to follow are incrementing, decrementing and choice. These models are illustrated for "m - n = ?" as follows:

1. Incrementing: Set a counter to n and count up until you reach m. For 6 - 4, the child starts with 4, and then counts 5, 6, and while counting extends one and then two fingers.

2. Decrementing: Set a counter to m and then count down n times. For 6 - 4, the child starts with 6 fingers extended, and then counts down 5, 4, 3, 2, while curling in 4 fingers. Two fingers are left extended as the difference.

3. Choice: Use either the incrementing or decrementing model depending on which requires the lesser amount of counting.

Their study yielded results similar to those in the Fuson study regarding the development of more sophisticated algorithms. The design of this study also was observational and data were from a small number of subjects.

Brown and Burton (1978) analyzed error patterns in learners' performance while learning three-column subtraction. The researchers were attempting to find if students had systematic "bugs" in the algorithms used in learning subtraction, rather than errors resulting from guessing. They used a computer program called BUGGY (Burton & Brown, 1979) to analyze the errors of 1,325 primary school students. About half of the students were found to have a bug or bugs in their subtraction algorithm. This suggested that systemic errors were widespread in subtraction among primary school students. Other errors may have been the result of guessing or bugs which BUGGY couldn't recognize.

Tatsuoka (1984, 1986) at the University of Illinois has developed rule-diagnostic models and computer programs for diagnosing student errors on the University's PLATO system. Tatsuoka also was concerned with examining changes in errors at different times and ways of designing remedial instruction.

Case (1978) found that algorithms must become automatic before more sophisticated algorithms can be used correctly, a finding that may have implications for the Groen and Parkman (1972), Fuson (1982), and Woods, et al. (1975) studies. Related work by Davis (1984) suggested expertise in computation involves being able to use algorithms automatically and correctly.

To conclude, the literature suggests research in the following areas related to the teaching of algorithms in problem solving:

1. Should students achieve high levels of automaticity on component problem solving skills before getting extensive training on more sophisticated algorithms?

2. Should students' performance on simple computation be analyzed in terms of the algorithm that is being used?

3. Should possible bugs be analyzed and should instruction be directed specifically at remediating the bugs?

The above section reviewed the literature related to the teaching of problem solving. Apparent productive approaches to looking at the teaching of problem solving were grouped into four categories: (a) translation training, (b) schema training, (c) strategy training, and (d) algorithm automaticity. These categories follow from concepts developed in associationism, Gestalt theory, and cognitive science. They offer the capability of studying problem solving and the effects of teaching problem solving in a less ambiguous way.

Literature on Problem Solving and Computers

The literature on problem solving and computers may be considered from three perspectives:

1. Problem solving environments
2. The use of artificial intelligence in problem solving

3. The impact of computer programming on problem solving

Problem Solving Environments

The first perspective on problem solving is related to the creation of environments to simulate or enhance problem solving. Many of the studies reported in this part of the review of the literature are qualitative and, therefore, have small numbers of subjects. The results and conclusions, however, tend to corroborate quantitative studies, including several of those discussed in the above section on teaching problem solving.

Papert (1980) and others at Massachusetts Institute of Technology developed a computer language called Logo. This language has been used to create environments for learning "powerful ideas," including problem solving. Logo has a set of graphics commands called turtle graphics. These commands can be used to move a "turtle" (cursor) around the screen. Triangles, squares, and other figures can be drawn by user programmed procedures.

Papert (1980) is very enthusiastic about Logo and problem solving. He suggests it can teach the processes of thinking and remove the need for formal classroom instruction:

> I see the classroom as an artificial and inefficient learning environment ... I believe that the computer's presence will enable us to so modify the learning environment outside the classroom that much if not all the knowledge schools presently try to teach with such

> pain and expense and such limited success will be learned ... without organized instruction. This obviously implies that schools as we know them today will have no place in the future. (pp. 31-2)

Papert's (1980) and other reports on Logo are largely anecdotal. His claims regarding the capability of Logo to enhance learning, thinking, and problem solving were not substantiated by objective research in this review of the literature. More recent literature has suggested the impact of Logo created environments may in fact have negative side effects (Teachers College, 1984). Studies in the last several years by Douglas Clements (1986), and Clements and Nastasi (in press) at State University of New York also have found that negative side effects may result from Logo environments.

Dennis and Kansky (1984) noted Papert's (1980) Logo as a means of providing students with experiences needed to teach mathematical thinking, but did not unconditionally accept Papert's claims about the Logo environment. They proposed that the computer is useful as a problem solving tool, helping the solver to manage the problem solving environment. They noted two practical examples of computer enhancements to problem solving: more efficient use of time and more ready access to large amounts of information.

Dennis and Kansky (1984) also suggested problem solving programs might be useful in developing problem solving abilities. These programs are "designed to aid the user in

exploring and solving some complex problem" (p. 191). They clearly stated that such programs were aids in solving complex problems, not a means to "crunch out" answers like a glorified calculator. In the problem solving environment suggested by Dennis and Kansky, a computer is a "means not an end. It serves rather than directs" (p. 192).

In an application which used a problem solving program, diSessa (1982) studied students learning Newtonian physics using Logo. He used a computer microworld called DYNA (for "dynamic turtle") to create an environment not possible without a computer. The graphics character on the screen (a turtle) moved according to the first two laws of Newton, the laws of inertia and force. The students could act upon the turtle by pressing R (right), L (left), or K (kick). R and L made the turtle turn in place 30 degrees. K applied a force of constant magnitude in the direction the turtle pointed. The turtle was shown moving vertically at a constant velocity. A target was to one side of the turtle's path. The student was to make the turtle hit the target.

Students almost always turned the turtle towards the target and kicked when it was directly across from the target- a strategy founded in Aristotelian, not Newtonian physics. diSessa (1982) reported that similar tasks modelled on the computer yielded similar results. The Logo environment enabled students to "unlearn" Aristotelian physics and learn Newtonian principles. It should be noted

that precise data were not given in diSessa's report of the study.

Subsequent work by diSessa (1983) has suggested a theory of developmental epistemology of physics misconceptions may be useful in understanding and remediating these errors. This theory is comprised of "phenomenological primitives" which shape one's view of the physical world. These primitives are the result of empirical or instructional experience. diSessa sees Logo as a means of shaping these primitives.

Related work on naive beliefs in physics has been done by McCloskey (1983). In his work, however, McCloskey only attempted to describe these misconceptions, not remediate them with Logo as diSessa did.

Heid (1983) reported on the use of muMath, a symbolic manipulator, in introductory calculus courses. In her article, Heid claimed students were freed of complex calculations and excelled in "higher level" cognitive skills, such as problem formulation and interpretation.

Moser and Carpenter (1982) studied problem solving among first-graders. Using a computer to model a solution for addition or subtraction problems, the child could add or remove counters on a screen by entering a complete number sentence. The child could use the computer to mimic a solution to an arithmetic problem with physical objects (e.g., houses) by creating a display of counters, then adding or subtracting them one at a time. The computer was

to be used to provide an environment in which students could develop semantics for number sentences. Moser and Carpenter reported that three of the four students used in their pilot study were able to transfer their computer abilities to traditional problems. Though the number of subjects was extremely low, the results appear consistent with the results of Mayer's (1983) translation training discussed earlier.

Little (1987) studied the effect of a writing environment created by Story Tree (Scholastic Inc., 1984) on student reading comprehension and vocabulary. The model used for reading comprehension involved the cognitive science concept of schemata discussed earlier. The software engaged students in schematic training and translation training to enhance the problem solving abilities needed to develop stories. Though no significant differences were reported between control and experimental groups, this study demonstrated an application of Mayer's (1983) problem solving teaching strategies to a non-mathematical subject. The study also suggested that a long period of treatment may be required to achieve significant differences between groups from such teaching strategies.

Artificial Intelligence

Much of the problem solving research carried out in artificial intelligence has been in the development of "expert systems." Machines are programmed to perform within a limited environment, mimicking the problem solving

behavior of a human expert (Barr & Feigenbaum, 1981; Michie, 1979).

Among the capabilities of these expert systems are the diagnosis of diseases, location of mineral deposits, and the decoding of the molecular structure of unknown compounds. Domains are limited and strictly defined, but these systems have been quite successful as practical applications of machine problem solving. Burton and Brown (1979) suggested that expert systems and other research in artificial intelligence may demonstrate useful ways of thinking about human problem solving and the enhancement of problem solving skills.

Mayer (1983) listed a variety of other problem solving activities related to the concerns of this study that are carried out using computers. These included: solving algebra story problems, solving geometry problems, generating logical proofs, playing chess, playing checkers, solving analogy problems, general problem solving, solving logical and deductive problems, understanding problem descriptions, teaching computer-programming strategies. To the extent that these activities model human problem solving activities, they may offer productive directions for further research in human problem solving.

Problem Solving and Computer Programming

Another focus of research in the literature on problem solving is the relationship between learning to program and learning problem solving. Mayer, et al. (1986) cited the

claims of Papert (1980) and Bork (1981) about computer programming and the development of problem solving skills. Papert suggested that when students learn to write programs in Logo, powerful intellectual skills are developed in the process. Bork saw computer programming as a vehicle for training in analytical thinking applicable to broad classes of problems. Mayer recalled similar claims of transfer of learning by the Latin School movement, which Thorndike (1923) suggested were unfounded, and suggested caution in making such claims.

More recently, Mayer, et al. (1986) studied the transfer of problem skills from computer programming. They defined four general sets of thinking skills involved in problem solving. These four sets were further subdivided. The groups and subgroups were:

1. Problem translation skills
    - Word problem translation
    - Word problem solution
2. Procedure comprehension skills
    - Following procedures
    - Following directions
3. General ability
    - Logical reasoning
    - Visual ability
    - Verbal ability
4. Other skills
    - Arithmetic computation

In their study, two groups of students were given pre- and posttests to measure these skills. One group took a course in BASIC programming, the other did not. Students were matched between groups by pretest scores, so that mean test scores on each pretest were identical for the two groups. Gain scores were computed after the posttest. The BASIC group was found to have gained significantly on two problem solving skills: problem translation and procedure comprehension. The BASIC group did not show significant gains on the tests of general intellectual ability and computation.

Soloway (1986) carried out a similar study of problem translation ability among university engineering students learning BASIC. The group of students who learned BASIC showed significant gains in problem translation ability.

Taitt (1985) investigated the effect of BASIC computer instruction on the problem solving abilities of pre-service elementary teachers. The control group was registered in "Teaching Elementary Mathematics" and was instructed in a geometry unit during the study. The experimental group was enrolled in "Microcomputers for Teachers" and was instructed in BASIC during the study. The pre- and posttests were comprised of multiple choice word problems and puzzles. The pre-service teachers who received the BASIC programming instruction significantly increased their problem solving ability.

In a subsequent study, Rucinski (1986) investigated the effect of computer programming on the problem solving strategies of both elementary and secondary pre-service teachers. The experimental groups were enrolled in "Microcomputers in Education" and taught BASIC programming. The control groups were enrolled in "Metric Geometry and Teaching Elementary Mathematics" and "Instructional Tasks in the Secondary School." The pretest and posttest were derived from problem solving situations developed at the Loyola University Psychometric Laboratory in the 1960's. The experimental group did not show significant gains in the strategies used in solving problems in the study. Rucinski suggested a follow-up study might be productive if similar students were to take more than one course in computer programming or in a language other than BASIC.

In a subsequent study, McGrath (1988) compared the problem solving skills of students learning a first or second programming language and those receiving no programming instruction. Programming languages were BASIC or Pascal. Half of each programming group also received a two-week problem solving intervention. The study was to determine whether instruction in a second language helped produce a transfer of problem solving skills. Six problem solving strategies were part of the problem solving intervention:

1. Use of reuseable procedures
2. Solution of problems by analogy

3. Use of recursive procedures
4. Creative problem solving
5. Use of the biconditional
6. Debugging

Students learning a second language were significantly better at debugging than those learning only a first language. McGrath (1988) suggested that further study is needed as to the effects of BASIC or Pascal as a first programming language. It was unclear if the gains in debugging were the result of subjects having a second language or having taken Pascal. She also suggested comparing the gains in problem solving skills of students taught programming with the gains of students taught only problem solving skills.

Tu (1988) studied the effects of taking BASIC, Pascal, or FORTRAN on the problem solving abilities of university students. Students were non-computer science majors enrolled in a semester programming course. Her problem solving definition and test instrument were the same as those used by Mayer, et al (1986). The pre- and posttests were presented on PLATO, a computer instructional system developed at the University of Illinois.

Results showed improvement of student problem solving abilities at the .05 level of significance. The logical reasoning component of problem solving improved the most. The Pascal group also had a significant improvement in the ability to solve word problems.

Tu (1988) recommended further study of the teaching methodologies used in the programming courses to compare their effectiveness in teaching programming. She also recommended a replication of her study using computer science majors. Tu suggested a study of users who have had computer programming to see if such experience is helpful in using computer software more productively.

Funkhouser (1990) investigated the effects of so-called "problem solving computer software" on the problem solving ability and attitudes toward mathematics of secondary school students. A cognitive science-based definition, instrumentation, and instructional pedagogy were used. Members of an experimental group were involved in computer-augmented mathematics instruction, while members of a control group were involved in more traditional mathematics instruction. Although results of a statistical analysis were mixed, the experimental group demonstrated significantly better performance on standardized tests of mathematics content. Students who used computer software also demonstrated positive gains in attitudes toward mathematics. A noteworthy example of these attitude gains was an increased perception by experimental students of mathematics as a gender-neutral domain. This perception increased more strongly for males than females.

Summary

Lester and Garofalo (1982) noted that much of the literature pertaining to the teaching of problem solving,

> is still largely based on folklore and the sage advice of teachers like George Polya, and can be characterized as being grounded more in the intuition of good teachers. (p. 26)

This review of the literature has attempted to give an overview of the research on problem solving, the teaching of problem solving, and the uses of computers to teach problem solving. The body of literature pertaining to problem solving is extensive. Taitt's (1985) study reported almost 15,000 listings produced by mathematics educators and psychologists related to problem solving. An ERIC search by Rucinski (1986) produced almost 10,000 hits for the descriptor "problem solving." This author's ERIC search and first-hand searches at various libraries also produced over 10,000 citations on problem solving and related literature. The literature I have discussed in this review is representative of that body of literature. It includes the most credible research done in this field. The literature selected suggests productive directions for future research into the process of problem solving and instruction in mathematical problem solving.

REFERENCES

Barr, A., & Feigenbaum, E. A. (1981). The Handbook of Artificial Intelligence (Vols. I & II). Los Altos, CA: William Kaufman, Inc.

Bartlett, F. C. (1932). Remembering: A study in experimental and social psychology. London: Cambridge University Press.

Bloom, B. S., & Broder, L. J. (1950). Problem-solving processes of college students. Chicago: University of Chicago Press.

Bork, A. (1981). Learning with computers. Bedford, MA: Digital Press, 1981.

Brown, J. S., & Burton, R. R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. Cognitive Science, 2, 155-92.

Bruner, J. (1961). The art of discovery. Harvard Educational Review, 31, 21-32.

Burton, R. R., & Brown, J. S. (1979). An investigation of computer coaching for informal learning activities. International Journal for Man-Machine Studies, 11, 5-24.

Case, R. (1978). Intellectual development from birth to adulthood: A neo-Piagetian interpretation. In R. S. Siegel (Ed.), Children's thinking: What develops? (pp. 76-89). Hillsdale, NJ: Lawrence Erlbaum Associates.

Chisko, A. M., & Davis, L. K. (1986). The analytical connection: Problem solving across the curriculum. Mathematics Teacher, 79, 592-596.

Clements, D. H. (1986). Effects of Logo and CAI environments on cognition and creativity. Journal of Educational Psychology, 78, 309-318.

Clements, D. H., & Nastasi, B. K. (in press). Social and cognitive interactions in educational computer environments. American Educational Research Journal.

Corman, B. R. (1957). The effect of varying amounts and kinds of information as guidance in problem solving. Psychological Monographs, 71, Whole No. 431.

Davis, R. B. (1984). Learning mathematics: The cognitive science approach to mathematics education. Norwood, NJ: Ablex Publishing Corporation.

Dennis, J. R., & Kansky, R. F. (1984). Instructional computing. Glenview, IL: Scott Foresman and Co.

diSessa, A. (1982) Unlearning Aristotelian physics: A study of knowledge-based learning. Cognitive Science, 6, 37-76.

diSessa, A. (1983). Phenomenology and the evolution of intuitions. In D. Gentner & A. L. Stevens (Eds.), Mental models (pp. 157-173). Hillsdale, NJ: Lawrence Erlbaum Associates.

Fuson, K. C. (1982). An analysis of the counting-on solution procedure in addition. In T. P. Carpenter, J. M. Moser, & T. A. Romberg (Eds.), Addition and subtraction: A cognitive perspective (pp. 121-130). Hillsdale, NJ: Lawrence Erlbaum Associates.

Greeno, J. G. (1978). A study of problem solving. In R. Glaser (Ed.), Advances in instructional psychology (Vol. 1) (pp. 73-80). Hillsdale, NJ: Lawrence Erlbaum Associates.

Greeno, J. G. (1980). Some examples of cognitive task analysis with instructional implications. In R. E. Snow, P. Federico, & W. E. Montague (Eds.), Aptitude, learning, and instruction (Vol. 2) (pp. 172-194). Hillsdale, NJ: Lawrence Erlbaum Associates.

Groen, G. J., & Parkman, J. M. (1972). A chronometric analysis of simple addition. Psychological Review, 97, 329-43.

Hayes, J. R., Waterman, D. A., & Robinson, C. S. (1977). Identifying relevant aspects of a problem text. Cognitive Science, 1, 297-313.

Heid, M. (1983). Calculus with muMath: Implications for curriculum reform. Computing Teacher, 11, 46-49.

Hilgard, E. R., Atkinson, R. C., & Atkinson, R. L. (1971). Introduction to psychology. New York: Harcourt, Brace, Jovanovich.

Hinsley, D., Hayes, Jr. R., & Simon, H. A. (1977). From words to equations. In P. Carpenter & M. Just (Eds.), Cognitive processes in comprehensions (pp. 136-150). Hillsdale, NJ: Lawrence Erlbaum Associates.

Funkhouser, C. P. (1990). The effects of problem solving software on the problem solving ability of secondary school students. Unpublished doctoral dissertation, University of Illinois, Urbana.

Kendler, T. S., and Kendler, H. H. (1962). Vertical and horizontal processes in problem solving. Psychological Review, 69, 1-16.

Kilpatrick, J. (1985). A retrospective account of the past 25 years of research on teaching mathematical problem solving. In E. A. Silver (Ed.), Teaching and learning mathematical problem solving: Multiple research perspectives (pp. 1-16). Hillsdale, NJ: Lawrence Erlbaum Associates.

Kohler, W. (1925). The mentality of apes. New York: Harcourt Brace Jovanovich.

Lester, F. K. & Garofalo, J. (1982). Introduction. In F. K. Lester & J. Garofalo (Eds.), Mathematical problem solving issues in research (pp. 1-11). Philadelphia: The Franklin Institute Press.

Little, D. (1987). The effects of story-building software on reading comprehension of middle school students (Doctoral dissertation, University of Illinois, Urbana). Dissertation Abstracts International, 48, 1804A.

Mayer, R. E. (1981). Frequency norms and structural analysis of algebraic story problems into families, categories, and templates. Instructional Science, 10, 135-75.

Mayer, R. E. (1982a). Different problem solving strategies for algebra word and equation problems. Journal of Experimental Psychology: Learning, Memory and Cognition, 8, 448-62.

Mayer, R. E. (1982b). Memory for algebra story problems. Journal of Educational Psychology, 74, 199-216.

Mayer, R. E. (1983). Thinking, problem solving, cognition. New York, NY: W. H. Freeman and Company.

Mayer, R. E. (1985). Implications of cognitive psychology for instruction in mathematical problem solving. In E. A. Silver (Ed.), Teaching and learning mathematical problem solving: multiple research perspectives (pp. 123-138). Hillsdale, NJ: Lawrence Erlbaum Associates.

Mayer, R. E., Dyck, J. L., & Vilberg, W. (1986). Learning to program and learning to think: What's the connection? Communications of the Association for Computing Machinery, 29, 605-610.

McClosky, M. (1983, April). Intuitive physics. Scientific American, 122-30.

McGrath, D. (1988). Problem solving and programming: Effects of a problem solving intervention in first and second programming languages. Unpublished manuscript.

Michie, D. (Ed.), (1979). Expert systems in the micro-electronic age. Edinburgh: University of Edinburgh Press.

Moser, J., & Carpenter, T. (1982). Using the microcomputer to teach problem solving skills: Program development and initial pilot study. Unpublished manuscript.

National Council of Teachers of Mathematics. (1926). Second yearbook. New York: Bureau of Publications, Teachers College, Columbia University.

National Council of Teachers of Mathematics. (1928). Third yearbook. New York: Bureau of Publications, Teachers College, Columbia University.

National Council of Teachers of Mathematics. (1980). An Agenda for action: Recommendations for School Mathematics for the 1980's. Reston, VA: National Council of Teachers of Mathematics.

Paige, J. M., & Simon, H. A. (1966). Cognitive processes in solving algebra word problems. In B. Kleinmetz (Ed.), Problem solving: Research, method, and theory (pp. 103-121). New York: John Wiley & Sons.

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. New York: Basic Books.

Polya, G. (1957). How to solve it. Garden City, New York: Doubleday Anchor.

Polya, G. (1965). Mathematical discovery (Vol. 2). New York: John Wiley & Sons.

Reif, F. (1981). Theoretical and educational concepts with problem solving: Bridging the gaps with human cognitive engineering. In D. T. Tuma & F. Reif (Eds.), Problem solving and education: Issues in teaching and research (pp. 144-162). Hillsdale, NJ: Lawrence Erlbaum Associates.

Resnick, L. B., & Ford, W. (1981). The psychology of mathematics for instruction. Hillsdale, NJ: Lawrence Erlbaum Associates.

Riley, M. (1982). The development of children's problem solving ability in arithmetic. In H. P. Ginsburg (Ed.), The development of mathematical thinking (pp. 23-35). New York: Academic Press.

Rubinstein, M. F. (1981). A decade of experience in teaching an interdisciplinary problem-solving course. In D. T. Tuma & F. Reif (Eds.), Problem solving and education: Issues in teaching and research (pp. 130-142). Hillsdale, NJ: Lawrence Erlbaum Associates.

Rucinski, T. T. (1986). The effect of computer programming on the problem solving strategies of preservice teachers (Doctoral dissertation, University of Illinois, Urbana). Dissertation Abstracts International, 47, 3404A.

Scholastic Inc. (1984) Story tree [Computer program]. New York: Scholastic Inc.

Schoenfeld, A. H. (1979). Explicitly heuristic training as a variable in problem solving performance. Journal for Research in Mathematics Education, 10, 173-87.

Schoenfeld, A. H. (1982). Some thoughts on problem solving research and mathematics education. In F. K. Lester & J. Garofalo (Eds.), Mathematical problem solving: Issues in research (pp. 1-11). Philadelphia: The Franklin Institute Press.

Silver, E. A. (1979). Students' perception of relatedness among mathematical verbal problems. Journal for Research in Mathematics Education, 10, 195-210.

Soloway, E. (1986). Learning to program = Learning to construct mechanisms and explanations. Communications of the Association for Computing Machinery, 29, 850-858.

Soloway, E., Lochhead, J., & Clement, J. (1982). Does computer programming enhance problem solving ability? Some positive evidence on algebra word problems. In R. J. Seidel, R. E. Anderson, & B. Hunter (Eds.), Computer literacy (pp. 321-335). New York: Academic Press.

Taitt, N. (1985). The effect of computer programming instruction on the problem solving ability of pre-service elementary teachers. (Doctoral dissertation, University of Illinois, 1985). Dissertation Abstracts International, 46, 1908A.

Tatsuoka, K. (1984). Changes in error types over learning stages. Journal of Educational Psychology, 76, 120-129.

Tatsuoka, K. (1986). Diagnosing cognitive errors: Statistical pattern classification based on item response theory. Behaviormetrika, 19, 73-86.

Teachers College. (1984). Teachers College Record, 85. New York: Columbia University, 539-639.

Thorndike, E. L. (1911). Animal intelligence. New York: Macmillan.

Thorndike, E. L. (1922). The psychology of arithmetic. New York: Macmillan.

Thorndike, E. L. (1923). The influence of first-year Latin upon the ability to read English. Schools and Society, 17, 165-168.

Tu, J. J. (1988). The relationship between learning to program and problem solving ability. Unpublished doctoral dissertation, University of Illinois, Urbana.

Underwood, B. J. (1965). False recognition produced by implicit verbal responses. Journal of Experimental Psychology, 70, 122-129.

Wertheimer, M. (1959). Productive thinking. New York: Harper & Row.

Woods, S. S., Resnick, L. B., & Groen, G. J. (1975). An experimental test of five process models for subtraction. Journal of Educational Psychology, 67, 17-21.